

Politechnika Śląska
Wydział Matematyk Stosowanej
Kierunek Informatyka

Gliwice, 01.10.2020

Programowanie I
projekt zaliczeniowy
"Galeria zdjęć"

Tomasz Korczowski gr. lab. 1

1. Opis projektu.

Galeria pozwalająca na przeglądanie zdjęć na komputerze z uwzględnieniem poruszania się po folderach.

Napisana w oparciu o bibliotekę SFML.

2. Wymagania

- Poruszanie się po katalogach zawierających zdjęcia
- Wyświetlanie zawartości bieżącego folderu w postaci „mozaiki” zdjęć
- Podgląd wybranego zdjęcia
- Skalowanie interfejsu do rozmiaru okna

3. Przebieg realizacji

3.1 Dobór narzędzi

Do realizacji interfejsu aplikacji została wybrana biblioteka SFML, która pozwala łatwo utworzyć okno, wyświetlać grafikę i śledzić urządzenia wejściowe.

Nie jest to optymalny dobór narzędzia, ponieważ SFML jest projektowany z myślą o grach; nie jest przystosowany do tworzenia aplikacji użytkowych. Wiązało się to z dodatkową pracą nad zorganizowaniem i rozłożeniem elementów interfejsu.

Decyzja o takim doborze narzędzia była motywowana szukaniem wyzwania; okazją do eksperymentowania i poszerzania swojego doświadczenia przy nieoczywistym problemie.

3.2 Pliki i wykonane zadania

Kod programu składa się z 23 plików liczących łącznie około 1200 lini kodu.

Pliki pomocnicze:

Te pliki powstały w celu zdefiniowania skrótów dla często używanych poleceń oraz implementacji często wykonywanych, żmudnych obliczeń (głównie związanych z przeliczaniem

współrzędnych i skalowałem). Częściowo służą dostosowaniu interfejsu SFML do potrzeb aplikacji. Znacznie ułatwiły prace nad pozostałymi funkcjonalnościami.

- Nodiscard.hpp
- filesystem.hpp
- Rect.hpp
- Rect.inl
- Vector.hpp
- Vector.inl
- SizedTexture.hpp
- SizedTexture.cpp
- fitTextInBox.hpp
- fitTextInBox.cpp
- SubTarget.pp
- SubTarget.cpp

Szablony Rect i Vector2 rozszerzają interfejsy szablonów sf::Rect i sf::Vector2 m.in. o metody pozwalające odczytać pozycję i rozmiar prostokąta. Było to ważne, ponieważ układ interfejsu aplikacji był definiowany jako prostokątne wycinki ekranu (klasa Subtarget).

Pliki fitTextInBox i SizedTexture pomagają w odpowiednim dopasowaniu tekstu i tekstury do obszaru ekranu na którym mają się znaleźć.

Pliki z głównymi elementami logiki aplikacji

Aplikacja służy do przeglądania grafik i folderów w związku z czym główne pliki to:

- Directory.hpp
- Directory.cpp
- Image.hpp
- Image.cpp

Klasa Image odpowiada za odpowiednie wczytanie i przechowanie grafik.

Klasa Directory przedstawia katalog w kontekście aplikacji. Klasa ta „widzi” tylko pliki graficzne oraz foldery, które takowe zawierają. Pozostałe pliki i foldery są pomijane. Odfiltruje również pliki ukryte (poprzedzone kropką).

Pliki interfejsu

Większą część kodu tej aplikacji stanowi jej interfejs.

- ImageGallery.hpp
- ImageGallery.cpp
- ImageViewer.hpp
- ImageViewer.cpp
- Button.hpp
- Button.cpp

Klasa Button reprezentuje prostokątny przycisk z ikoną i tekstową etykietą. Po wciśnięciu wywołuje zadaną mu funkcję.

Klasa ImageViewer wyświetla w powiększoną grafikę. Służy oglądaniu zdjęć.

Klasa ImageGallery jest najbardziej złożona. Wyświetla wszystkie grafiki obecne w folderze w postaci mozaiki „kafelków”. Pozwala też na przewijanie galerii i zaznaczenie zdjęcia (w celu wyświetlenia jego większej wersji w ImageViewer). Rozporządza ilością kolumn starając się optymalnie wykorzystać dostępną przestrzeń.

Plik main.cpp

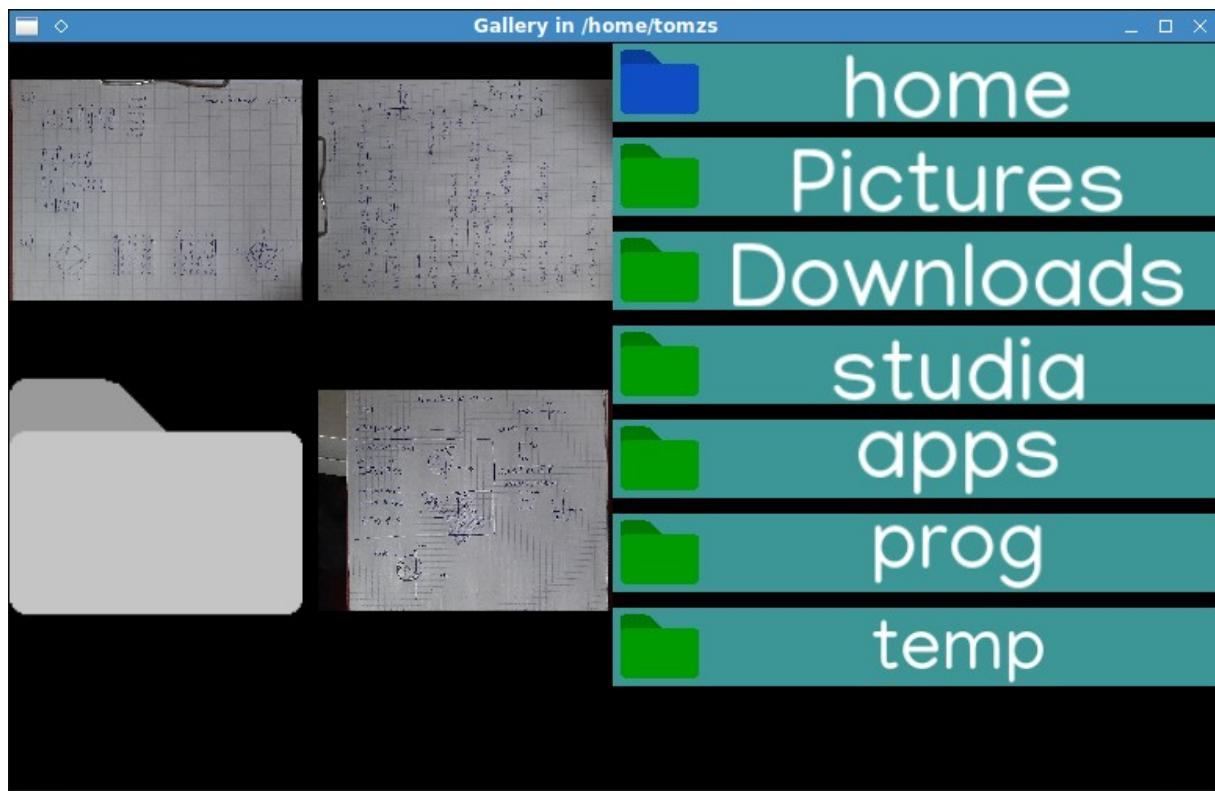
Ten plik składa projekt w całość; służy jako komunikacja między elementami interfejsu i definiuje ich pozycje i wymiary. Tworzy okno. Przekazuje elementom interfejsu otrzymany od użytkownika input.

Aby zachować skalowalność interfejsu wymiary i pozycje elementów zostały zaimplementowane jako wyrażenia lambda, aby łatwo móc je obliczać na nowo gdy interfejs ulegnie zmianie (np. w przypadku zmiany rozmiaru okna aplikacji).

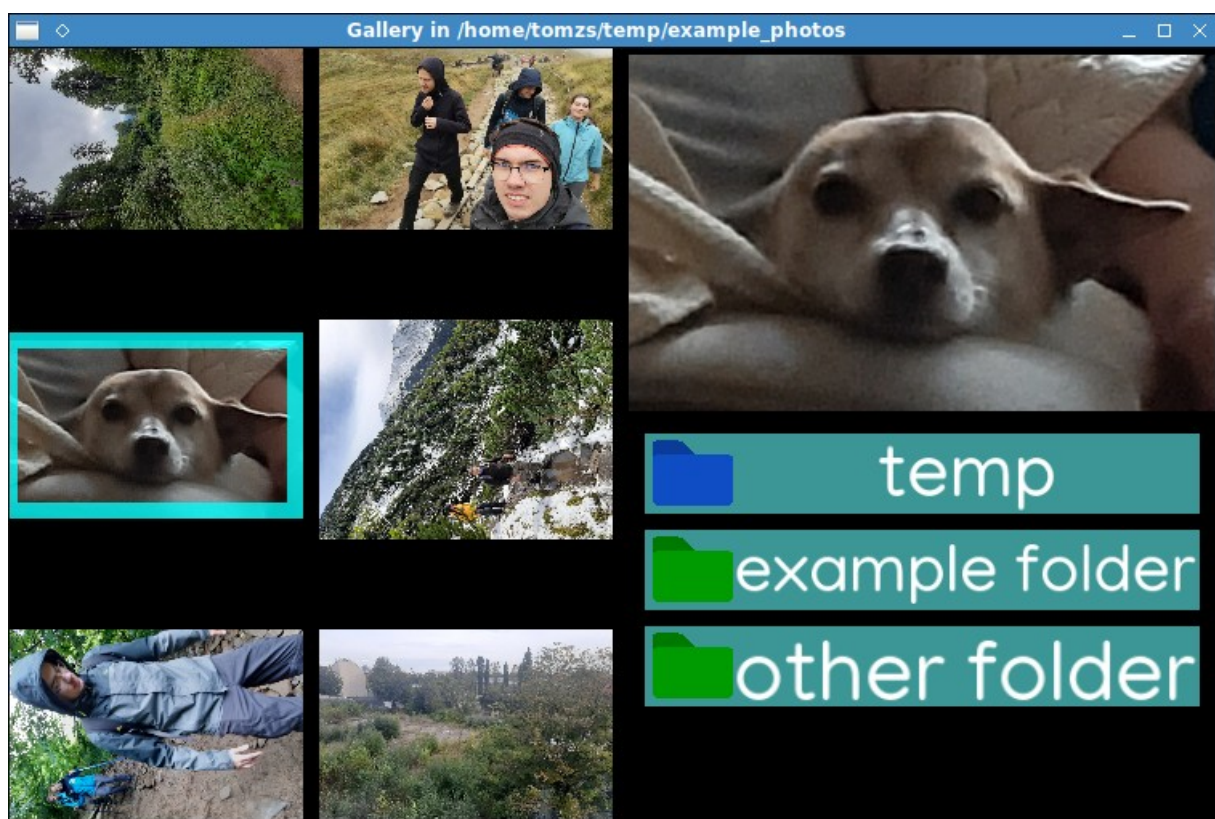
4. Instrukcja użytkownika

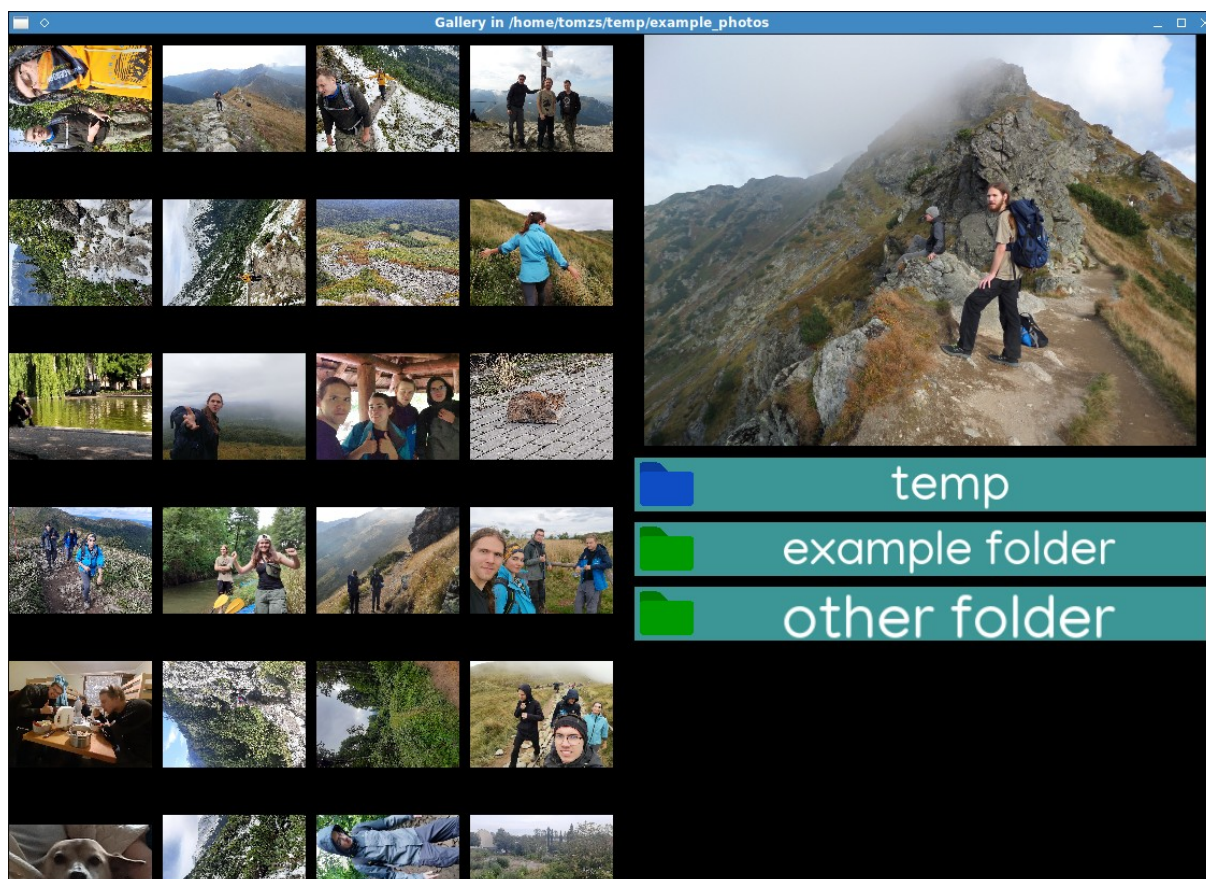
Program domyślnie otwiera się w głównym folderze użytkownika. Inny początkowy folder można też wybrać podając jego ścieżkę jako argument konsolowy.

Po lewej stronie okna wyświetlają się grafiki obecne w folderze, po prawej przyciski kierujące do dostępnych folderów. Pierwszy z nich (wyróżniający się niebieską ikoną) prowadzi do folderu



nadrzędnego. Po kliknięciu w grafikę z lewego panelu pokaże się ona w powiększeniu po prawej stronie ekranu, nad folderami.





5. Podsumowanie i wnioski.

Realizację projektu rozpocząłem dość późno ze względu na długi czas wachania przy wyborze tematu. Nie uwzględniłem też, że pod koniec semestru w natłoku kolokwiów z innych przedmiotów trudno jest wygospodarować odpowiednią ilość czasu dla projektu. Decyzja pisania interfejsu z użyciem jedynie SFML okazała się w tych warunkach kosztowna.

Według pierwotnego zamysłu program miał służyć organizowaniu zdjęć poprzez przypisywanie im etykiet, jednak realizacja bardziej podstawowych elementów pochłonęła zbyt dużo czasu.

W trakcie produkcji natrafiłem również na problem, którego niestety nie zdążyłem naprawić. Gdy program otwiera folder, interfejs zatrzymuje się do czasu załadowania wszystkich obecnych w nim grafik. Jeśli folder zawiera bardzo dużą ilość dużych zdjęć powoduje to widoczne zawieszenie się programu,

w skrajnych przypadkach prowadząc do jego awaryjnego zamknięcia.

Planowałem rozwiązać to przez przeniesienie odczytu z dysku do innego wątku, jednak niestety okazało się to nieosiągalne w wyznaczonym terminie. Zaczęte próby są w osobnej gałęzi repozytorium.

Program był pisany, kompilowany i testowany na systemie Linux. Nie używa jednak żadnych poleceń specyficznych dla tego systemu, więc potencjalny port na inne systemy powinien być osiągalny przez wprowadzenie zmian jedynie w linkowaniu.